SCIENCES
SORBONNE
UNIVERSITÉ

TP 3E204

2020 - 2021

TP1 de MC ARM7

Entrée/Sortie

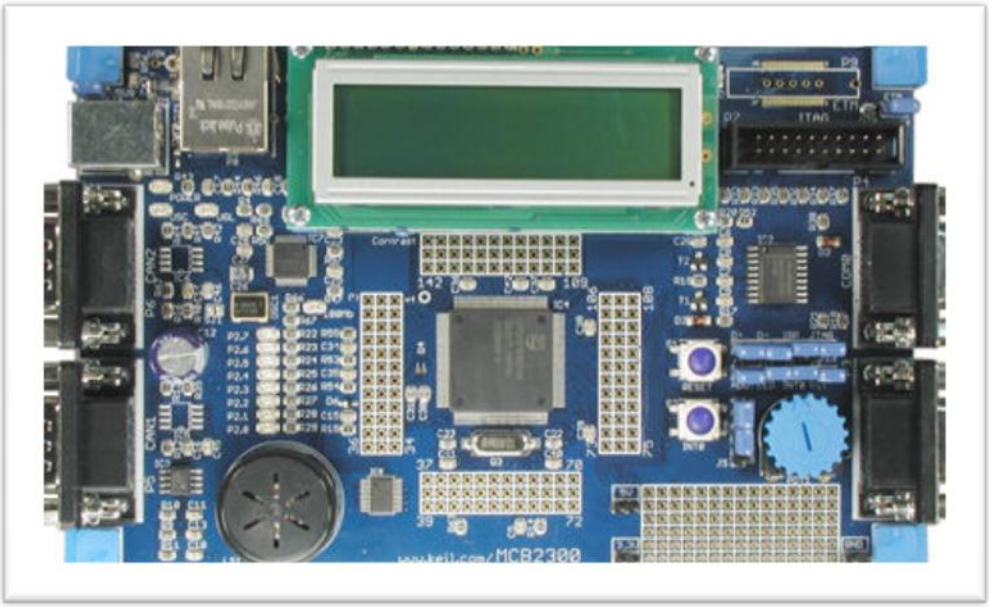


Figure 1 : Carte MCB2300

Les TP seront réalisés sur la carte de développement MCB2300 de la société KEIL. Cette carte s'articule autour d'un microcontrôleur NXP LPC2378. Ce circuit comprend un cœur ARM7TDMI, 32 ko de RAM statique, 512 ko de mémoire Flash, et un certain nombre de périphériques (USB, Ethernet, Bus CAN, Timers, interface I2C, convertisseurs A/N, UARTs…) lui permettant de piloter des composants externes. Il est cadencé à l'aide d'un oscillateur externe à la fréquence de 12 MHz.
La carte dispose en outre de plusieurs périphériques connectés au microcontrôleur : 2 ports série, 2 ports CAN, 1 interface Ethernet, 1 interface USB, 1 interface SD Card, 8 LEDs, 1 afficheur LCD, 2 boutons poussoirs, 1 haut parleur et un potentiomètre. On trouvera en annexe un schéma de câblage de la carte. L'alimentation de la carte ainsi que la communication avec le PC hôte et environnement de développement s'effectue à l'aide d'une liaison USB.
L'environnement logiciel utilisé au cours de ces TP est ***Keil µvision 3***. Il permet d'écrire les programmes, de les compiler, puis de les simuler et de les télécharger sur la carte. La notice d'utilisation du logiciel µVision 3 peut vous aider à utiliser le logiciel.

TRAVAIL A EFFECTUER

Exercice 1 : Initiation au simulateur de KEIL :

Lire attentivement la notice en annexe 1 et suivez les différentes étapes pour se familiariser avec KEIL UVision.

EXERCICE 2 – Allumage des LEDs

Allez écrire un programme simple qui va consister à écrire la valeur 0x55 sur les LEDs connectés au port P2. Ecrire les fonctions void init_GPIO() et int main() qui vont permettre d'écrire 0x55 sur les LEDs. Ouvrez le projet model Blinky. Créer votre fichier test.c et ajouter le au projet Blinky. Compiler et charger l'application sur la carte. Tester le bon fonctionnement.

EXERCICE 3 – Clignotement des LEDs

Maintenant, on veut faire clignoter les LEDs. Mais attention, car la fréquence du CPU est trop grande par rapport à la persistance rétinienne. Il faut donc rajouter une boucle d'attente après chaque changement sur les LEDs, pour cela vous allez créer la fonction void delay() qui permet de générer ce temps d'attente. Modifier votre programme afin de réaliser cette nouvelle fonctionnalité et tester-la sur la carte.

EXERCICE 4

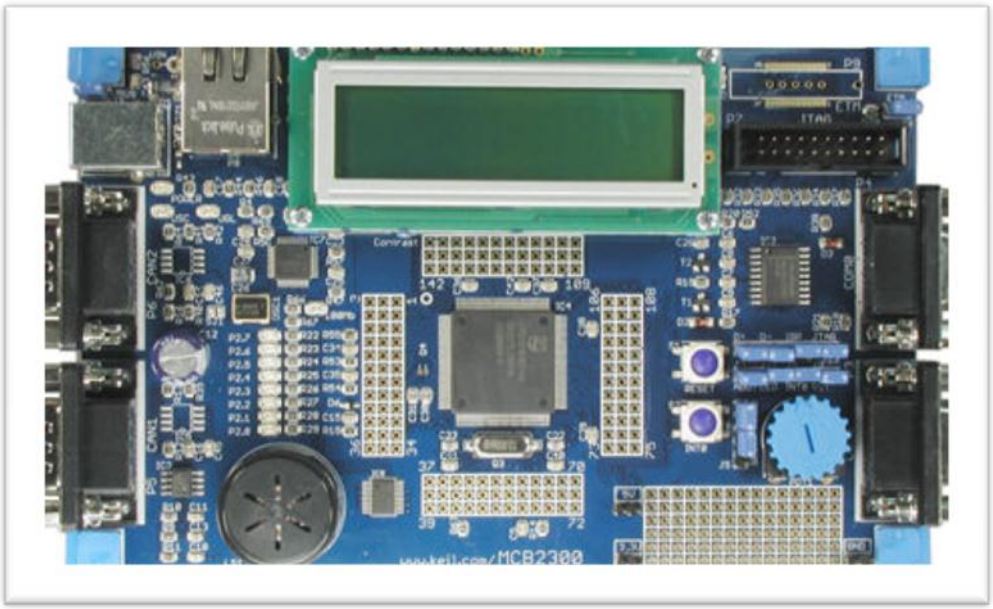
1. Ecrire un nouveau programme qui permet de lire la valeur qu'il y a sur les interrupteurs P4.0-3 et la recopie sur les LEDs (tester uniquement en Simulation).

2. Ecrire un programme qui génère un compteur modulo 200 sur les LEDs.

3. Ecrire un programme qui permet d'allumer les LEDs comme un chenillard, décalage de droite à gauche et ensuite de gauche à droite.

4. Rajouter un tempo pour que le décalage soit moins rapide.

TP 2 : Interruption Externe



EXCERCICE 1 Lecture du bouton poussoir par scrutation

On désire à présent faire clignoter les LEDs que lorsque le bouton poussoir est appuyé. Pour cela il faut scruter/tester le bouton poussoir en continu afin de connaitre son état à chaque instant (on parle alors de « polling »). Pour savoir avec quelle pin est connecté le bouton poussoir, il faut aller voir le schéma de câblage. Modifier votre programme afin de réaliser cette nouvelle fonctionnalité et tester-la sur la carte.

EXERCICE 2 – Lecture du bouton poussoir par interruption

Dans cet exercice, l'application doit allumer la LED reliée au port P2.0 lorsqu'on exerce une pression sur le bouton poussoir.

1. Quels sont les périphériques utilisés par l'application ?

2. Quels registres faut-il affecter pour configurer ces périphériques ?

3. Quelles valeurs faut-il affecter à ces registres ?

4. Pour la LED, écrire un sous-programme d'initialisation.

5. Pour le bouton poussoir, écrire un sous-programme d'initialisation et un sous-programme d'interruption.

6. Ecrire le programme principal qui initialise les périphériques et la gestion des interruptions, puis qui met le processeur en attente active.

7. Si vous avez accès à un environnement KEIL µVision3, compilez votre programme, construisez un exécutable et exécutez le en mode pas-à-pas en affichant le port P2. Vous pourrez simuler une pression sur le bouton poussoir en modifiant avec la souris la valeur de P2.10. Vérifier sur la carte le bon fonctionnement.

EXERCICE 3 – COMPTAGE DES PRESSIONS.

Dans cet exercice, l'application doit compter le nombre de pressions effectuées sur le bouton et l'afficher sur les 8 LEDs reliées aux ports P2.0 à P2.7.

1. Quelles modifications apporter aux programmes de l'exercice 2 pour cela ?

2. Si vous avez accès à un environnement KEIL µVision3, compilez votre programme, construisez un exécutable et exécutez le en mode pas-à-pas en affichant le port P2.

EXERCICE 4 – COMMUTATION DE LA LED

Dans cet exercice, l'application doit commuter la LED reliée au port P2.0 à chaque pression sur le bouton poussoir : si elle est éteinte, il faut l'allumer ; si elle est allumée, il faut l'éteindre.

1. Quelles modifications faut-il apporter aux programmes et sous-programmes de l'exercice 1 pour cela ?

2. Ecrire les programmes et sous-programmes répondant au cahier de charge.

3. Si vous avez accès à un environnement KEIL µVision3, compilez votre programme, construisez un exécutable et exécutez le en mode pas-à-pas en affichant le port P2. Vous pourrez simuler une pression sur le bouton poussoir en modifiant avec la souris la valeur de P2.10.

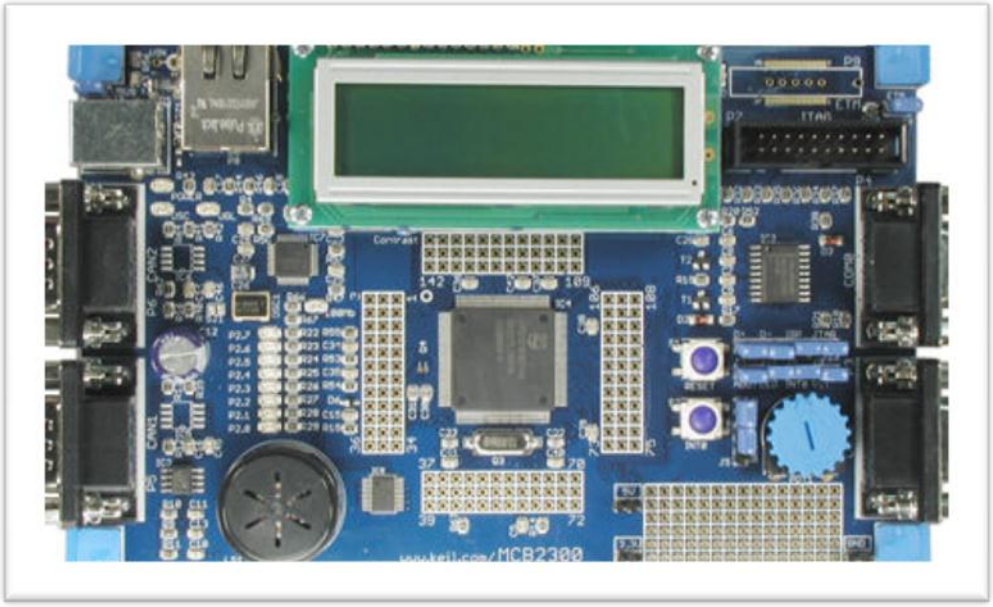
EXERCICE 5 – ALLUMAGE PROGRESSIF DES 8 LEDs

Dans cet exercice, l'application doit allumer successivement les 8 LEDs reliées aux ports P2.0 à P2.7. A chaque chaque pression sur le bouton poussoir, la LED suivante dans l'ordre de 0 à 7 est allumée. Si les 8 LEDs sont allumées, il faut toutes les éteindre.

1. Ecrire les programmes et sous-programmes répondant au cahier de charge.

2. Si vous avez accès à un environnement KEIL µVision3, compilez votre programme, construisez un exécutable et exécutez le en mode pas-à-pas en affichant le port P2. Vous pourrez simuler une pression sur le bouton poussoir en modifiant avec la souris la valeur de P2.10.

TP 3 : TIMERS



TRAVAIL A EFFECTUER

L'objectif de cette séance de TP est de mettre en œuvre les interruptions externes et les timers du LPC2378 pour faire clignoter les 8 LEDs.

EXERCICE 1 – Timers

On désire modifier de façon périodique l'état des LEDs de la carte. Chaque état des diodes (allumé / éteint) sera maintenu pendant une durée programmée dans un des timers du LPC2378. Le timer sera en outre utilisé pour générer une interruption périodique qui provoquera le changement d'état des diodes. On rappelle que le microcontrôleur est cadencé à une fréquence XTAL=12 MHz. Les périphériques du LPC378 fonctionnent quant à eux par défaut à une fréquence de 12 MHz.

1. *Ecrire un programme faisant clignoter les LEDs. La période de clignotement devra être de 500 ms, 1s, 2s et «10s à vérifier avec un chrono».*

2. *Programmer un compteur modulo 255 dont la valeur sera affichée sur les LEDs. Chaque valeur du compteur sera maintenue pendant 100 ms.*

EXERCICE 2 – Chenillard sans BP

On désire à présent générer une séquence sur les LEDs. La vitesse de défilement des LEDs sera programmée dans un des Timers du LPC2378. Les diodes devront s'allumer de gauche à droite et ensuite de droite à gauche de manière continu (comme un chenillard). La vitesse de défilement sera d'environ 200 ms. Programmer le LPC2378 pour réaliser cette nouvelle fonctionnalité.

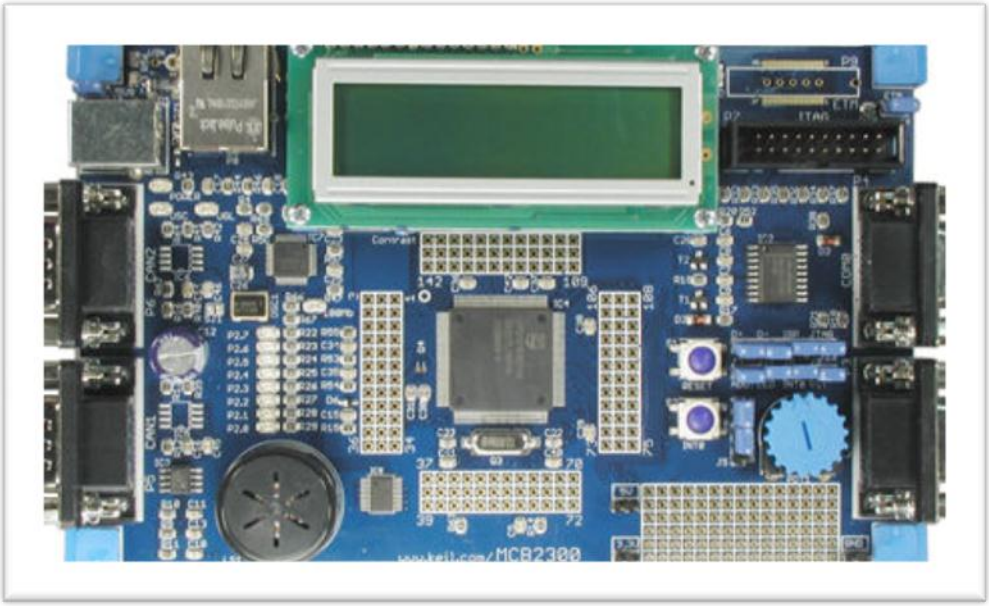
EXERCICE 3 – Chenillard avec BP

On désire à présent générer une séquence sur les LEDs lorsque on appui sur le bouton poussoir INT0. La vitesse de défilement des LEDs sera programmée dans un des Timers du LPC2378.

1. *Modifier le programme pour que le défilement des diodes commence lorsque on appui sur le bouton poussoir et s'arrête lorsque on appui à nouveau sur le bouton poussoir.*

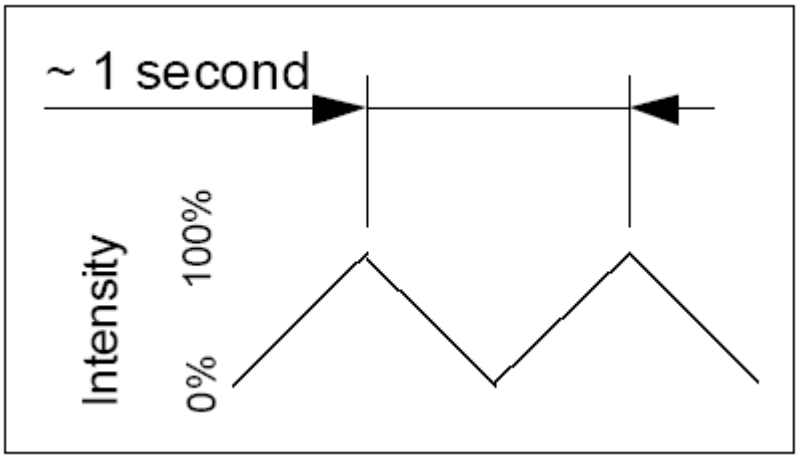
2. *Maintenant, on veut que après chaque appui sur le bouton poussoir, les diodes défilent une seule fois de gauche à droite et ensuite de droite à gauche. Au repos, la LED connecté sur P2.0 doit être allumé et les autres éteintes. Si on appui pendant que la séquence n'est pas terminée, le chenillard reprend au début de la séquence.*

TP 4 : PWM

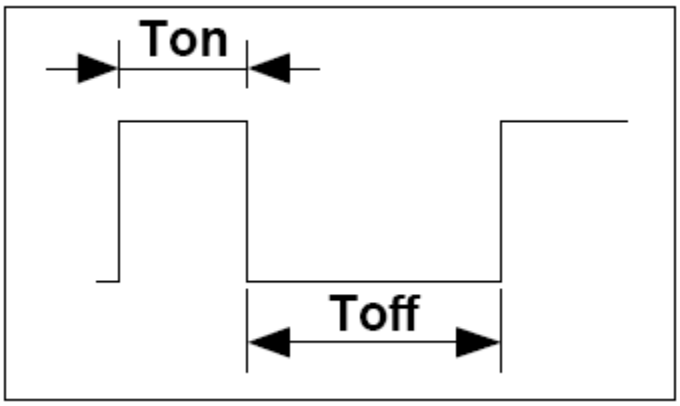


EXERCICE 1 – HeartBeat (10 pts)

Le but de cet exercice est d'afficher sur une des LEDs de la carte MCB2300 un « cœur qui bat» (*HeartBeat* en anglais). Pour cela, il faut moduler l'intensité perçue de la diode luminescente par une fonction triangulaire comme ci-dessous. Il n'est pas demandé une période très précise, mais une variation « autour » de 1 hertz (entre 0.5 et 2 Hz par exemple).



Pour cet effet de variation proportionnelle, nous utiliserons une technique dite Modulation de Largeur d'Impulsion, soit PWM en anglo-saxon (Pulse Width Modulation) représentée ci-dessous



L'intensité moyenne est donc proportionnelle au rapport cyclique **Ton / (Ton + Toff).** On peut donc faire varier **Ton** en gardant **Ton+Toff** constant. Pour que l'oeil ne voie pas les allumages et extinctions de la LED, il faut que la période Ton+Toff soit suffisamment faible. Une fréquence supérieure à 50 Hz (Ton+Toff < 20 ms) suffit. D'un autre côté, on évitera également les périodes trop courtes peu compatibles avec les temps de commutation de l'électronique de commande, et génératrices d'interférences. Un bon choix se situe entre 60 Hz (16.7 ms) et 400 Hz (2.5 ms).
Remarque : dans la pratique, l'intensité lumineuse perçue n'est pas une fonction linéaire, mais nous négligerons ceci pour cette application simple.

1. Commencer par écrire un programme qui permet de générer un signal PWM avec un rapport cyclique de 30 % sur le canal PWM1.1 qui est connecté à la LED0. Allumer les autres LEDs pour voir la différence d'intensité.

2. Modifier votre programme afin de faire clignoter cette LED (LED 0 ou P2.0) comme un coeur qui bat (HeartBeat). Pour cela, il faut changer le rapport cyclique du PWM périodiquement.
3. Faites la même chose pour les LEDs 1 à 5.

EXERCICE 2 – PWM commandé par BP (5 pts)

On désire pouvoir modifier l'intensité des LEDs à l'aide du bouton poussoir. Au démarrage du programme, les LEDs seront éteintes et à chaque appui sur le bouton poussoir l'intensité des LEDs devront augmenter de 10%. Lorsque l'intensité arrivera à 100%, elle sera réinitialisée à 0% (LEDs éteintes). Programmer le LPC2378 pour réaliser cette nouvelle fonctionnalité.

EXERCICE 3 – Vitesse du HeartBeat commandé par BP (5 pts)

Maintenant, on désire commander la vitesse de clignotement du HeartBeat avec un bouton poussoir. Au démarrage, faites clignoter le HeartBeat avec une période de 500 ms. A chaque appui sur le bouton poussoir la vitesse de clignotement doit augmenter de 200 ms. Lorsque la période de clignotement est supérieure à 2 secondes, réinitialisé là à 500 ms. Programmer le LPC2378 pour réaliser cette nouvelle fonctionnalité.